UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/388,766	09/02/1999	BOHR-WINN SHIH	303.513US1	4410

21186 7590 11/08/2002

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER	
SHARON, AYAL I	

ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 11/08/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/388,766	SHIH ET AL.	
	Examiner	Art Unit	
	Ayal I. Sharon	2123	

-- The MAILING DATE of this communication appears on the cov r sheet with the correspond nce address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *12 August 2002* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Introduction

1. Claims 1-35 of U.S. Application 09/388,766 filed on 09/02/1999 are presented for examination. Applicant's arguments filed 08/12/1999 have been fully considered but they are not persuasive.

Claim Interpretations

2. Examiner interprets a "node" as being an input or output to one or more gates in a circuit. Thus, Examiner interprets that the terms "signal" and "node" are interchangeable.
3. Examiner interprets a "forced" logic value as being one of the "forcing" values defined in the definition of the "std_logic" package in IEEE Standard 1164-1993, p.2, which is held constant.
4. Applicant defines the "release" of a node as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced." (Specification, p.5)
5. Examiner interprets the "release of a node" as being equivalent to enabling a change in the logic value of the node.

6. Applicant defines "resolving" a node as meaning that "the inputs to the node are known and therefore the logic value of the node can be determined through simulation by the simulation program." (Specification, pp.5-6)
7. Examiner interprets "resolving" a node as having its value determined according to the resolution table and truth tables defined in IEEE Standard 1164-1993, pp.4-5.
8. Examiner interprets that one possible "predetermined condition" for "releasing" a node is a rising or falling edge of a clock pulse.
9. Examiner interprets "predetermined amount of time" as being the length of an entire clock cycle or a number of clock cycles.
10. Examiner interprets that a waveform plot (signal graph) is a form of "error indication" and "providing an indication when the node is in an undesirable condition", and "outputting a node condition".
11. Examiner interprets that a user-defined length of a clock cycle, or a user-defined number of clock cycles, constitutes a "user-defined time period".
12. Examiner interprets "therefrom" as being equivalent to "from this" or "from that".
13. Examiner interprets "conveyance" as being the input of a signal value.

Claim Rejections - 35 USC § 102

14. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) do not apply to the examination of this application as the application being examined was not (1) filed on or after November 29, 2000, or (2) voluntarily published under 35 U.S.C. 122(b). Therefore, this application is examined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

15. The prior art cited is as follows:

16. IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164). Copyright 1993. (Henceforth referred to as "IEEE 1164").

17. Mueller, Martin. "Chronology TimingDesigner V1.2", Printed Circuit Design, San Francisco, CA. January 1993. (Henceforth referred to as "Mueller")

18. The claims are subsequently recited for Applicant's convenience. Applicant's attention is also directed to the pertinent sections of the prior art.

19. Claims 1-4, 6-8, 12-18, 20-24 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Mueller.

20. Claims 25-27 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by IEEE 1164.

21. Mueller teaches the limitations of Claim 1:

1. A method of simulating a node, comprising:
forcing an initial logic state on the node;
(Mueller, p.2, "Creating Signals")

releasing the node if a predetermined condition is met and creating therefrom a released node;
(Mueller, p.2, "Creating Signals", "Creating Clocks", and p.3 "Delays and Constraints", "Miscellaneous")

monitoring the released node; and
(Mueller, p.3, "Delays and Constraints", "Miscellaneous")

providing an indication when the released node is in a preselected condition.
(Mueller, p.3, "Miscellaneous")

22. As per Claim 2, Mueller teaches the limitations of Claim 1, as described above.

Mueller also teaches the limitations of Claim 2:

2. The method of claim 1, wherein forcing the initial logic state includes forcing to a logic zero, logic one or high-impedance.
(Mueller, p.2, "Creating Signals")

23. As per Claim 3, Mueller teaches the limitations of Claim 1, as described above.

Mueller also teaches the limitations of Claim 3:

3. The method of claim 1, wherein releasing the node further comprises determining that the condition is met after passage of a predetermined amount of time.
(Mueller, p.3, "Delays and Constraints")

24. As per Claim 4, Mueller teaches the limitations of Claims 1 and 3, as described above. Mueller also teaches the limitations of Claim 4:

4. The method of claim 3, wherein releasing the node further comprises determining that the condition is met when the node has been resolved.
(Mueller, p.2 "Creating Clocks", p.3, "Delays and Constraints", "Miscellaneous")

25. As per Claim 6, Mueller teaches the limitations of Claim 1, as described above.

Mueller also teaches the limitations of Claim 6:

6. The method of claim 1, further comprising providing an error indication when the released node is a preselected condition.
(Mueller, p.3, "Miscellaneous")

26. As per Claim 7, Mueller teaches the limitations of Claims 1 and 3, as described above. Mueller also teaches the limitations of Claim 7:

> 7. The method of claim 3, further comprising selecting a user-defined time period for the predetermined amount of time.
> (Mueller, p.2, "Design Environment", and p.3 "Delays and Constraints", and p.3 "Miscellaneous")

27. Mueller teaches the limitations of Claim 8:

> 8. A method of initializing and monitoring a simulated circuit node, comprising:
>
> obtaining an initial node condition for a node;
> (Mueller, p.2 "Creating Signals", and p.3 "Delays and Constraints", and p.3, "Miscellaneous")
>
> forcing the node to the initial node condition;
> (Mueller, p.2 "Creating Signals")
>
> simulating a circuit containing the node;
> (Mueller, p.2 "Creating Signals", and p.3 "Delays and Constraints", and p.3, "Miscellaneous")
>
> testing the node for a valid condition;
> (Mueller, p.2 "Creating Signals", and p.3, "Miscellaneous")
>
> monitoring the node; and
> (Mueller, p.3, "Miscellaneous")
>
> providing an indication when the node is in an undesirable condition.
> (Mueller, p.3, "Delays and Constraints", "Miscellaneous")

28. As per Claim 12, Mueller teaches the limitations of Claim 8, as described above. Mueller also teaches the limitations of Claim 12:

> 12. The method of claim 8, further comprising outputting the condition of the simulated node.
> (Mueller, p.3, "Delays and Constraints", "Miscellaneous")

29. As per Claim 13, Mueller teaches the limitations of Claim 8, as described above. Mueller also teaches the limitations of Claim 13:

> 13. The method of claim 8, further comprising obtaining a simulation run time.
> (Mueller, p.3, "Delays and Constraints", "Miscellaneous")

30. As per Claim 14, Mueller teaches the limitations of Claims 8 and 13, as described above. Mueller also teaches the limitations of Claim 14:

14. The method of claim 13, further comprising outputting a final node condition when the simulation run time is completed.
(Mueller, p.3, "Delays and Constraints", "Miscellaneous")

31. Mueller teaches the limitations of Claim 15:

15. A computer-readable medium having computer-executable instructions comprising:

forcing an initial logic state on the node;
(Mueller, p.2, "Creating Signals")

releasing the node if a predetermined condition is met and creating therefrom a released node;
(Mueller, p.2, "Creating Signals", "Creating Clocks", and p.3 "Delays and Constraints")

monitoring the released node; and
(Mueller, p.3, "Delays and Constraints", "Miscellaneous")

providing an indication when the released node is in a preselected condition.
(Mueller, p.3, "Miscellaneous")

32. As per Claim 16, Mueller teaches the limitations of Claim 15, as described above.

Mueller also teaches the limitations of Claim 16:

16. The medium of claim 15, having further computer-executable instructions for forcing the initial logic state to a logic zero, logic one or high-impedance.
(Mueller, p.2, "Creating Signals")

33. As per Claim 17, Mueller teaches the limitations of Claim 15, as described above.

Mueller also teaches the limitations of Claim 17:

17. The medium of claim 15, having further computer-executable instructions for determining that the condition is met after passage of a predetermined amount of time.
(Mueller, p.3, "Delays and Constraints")

34. As per Claim 18, Mueller teaches the limitations of Claim 15, as described above.

Mueller also teaches the limitations of Claim 18:

18. The medium of claim 15, having further computer-executable instructions for determining that the condition is met when the node has been resolved
(Mueller, p.2 "Creating Clocks", p.3, "Delays and Constraints", "Miscellaneous")

35. Mueller teaches the limitations of Claim 20:

20. A simulation module for initializing and monitoring a simulated circuit node, comprising:

an input means for inputting an initial node condition;
(Mueller, p.2, "Design Environment", "Creating Signals", p.3 "Miscellaneous")

a conveying means for conveying the initial node condition to a simulated node;
(Mueller, p.2, "Design Environment", "Creating Signals", p.3 "Miscellaneous")

release means for releasing the node upon satisfaction of a condition;
(Mueller, p.2, "Design Environment", "Creating Signals", p.3 "Delays and Constraints", "Miscellaneous")

a monitoring means for monitoring the simulated node for a node condition;
(Mueller, p.3, "Miscellaneous")

and an output means for outputting an indication when the node condition is in an undesirable state.
(Mueller, p.3, "Miscellaneous")

36. As per Claim 21, Mueller teaches the limitations of Claim 20, as described above.

Mueller also teaches the limitations of Claim 21:

21. The module of claim 20, further comprising an output means for outputting the node condition.
(Mueller, p.3, "Miscellaneous")

37. As per Claim 22, Mueller teaches the limitations of Claim 20, as described above.

Mueller also teaches the limitations of Claim 19:

22. The module of claim 20, further comprising an input means for inputting a simulation run time.
(Mueller, p.2-3, "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous")

38. As per Claim 23, Mueller teaches the limitations of Claim 22, as described above.

Mueller also teaches the limitations of Claim 23:

23. The module of claim 22, further comprising an output means for outputting a final node condition at completion of the simulation run time.
(Tzartzanis, pp.63, 67-70)

39. Mueller teaches the limitations of Claim 24:

24. A computerized system for initializing and monitoring a simulated circuit node, the system comprising:
a circuit simulation tool;
(Mueller, p.1, Abstract, and para. 1)

a first input module inputting an initial node condition;

(Mueller, p.2, "Creating Signals")

a conveying module conveying the initial node condition to a simulated node;
(Mueller, p.2-3, "Creating Signals", "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous")

a release module releasing the initial condition;
(Mueller, p.2-3, "Creating Signals", "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous")

a monitoring module monitoring the simulated node for a node condition;
(Mueller, p.2-3, "Creating Signals", "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous")

a first output module outputting an indication when the node condition is in an undesirable state;
(Mueller, p.2-3, "Creating Signals", "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous")

a second input module inputting a simulation run time; and
(Mueller, p.2-3, "Creating Signals", "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous" Note: Mueller expressly teaches "Each new signal" ... and "Printing of timing diagrams is done by specifying the range of signals and time intervals to be printed")

a second output module outputting a final node condition at completion of the simulation run time.
(Mueller, p.2-3, "Creating Signals", "Creating Clocks", p.3 "Delays and Constraints", "Miscellaneous")

40. IEEE 1164 teaches the limitations of Claim 25:

25. An HDL initial condition module comprising a means for maintaining a logic level of a simulated circuit node until a release condition is met.
(IEEE 1164, pp.2-5)

41. As per Claim 26, IEEE 1164 teaches the limitations of Claim 25, as described above. IEEE 1164 also teaches the limitations of Claim 26:

26. The module of claim 25 wherein the release condition is when the node can be resolved to a known logic state.
(IEEE 1164, pp.2-5)

42. As per Claim 27, IEEE 1164 teaches the limitations of Claim 25, as described above. IEEE 1164 also teaches the limitations of Claim 27:

27. The module of claim 25 wherein the logic level is a value defined by an HDL executable simulation program.

(IEEE 1164, pp.2-5)

VHDL is an HDL executable simulation language.

Claim Rejections - 35 USC § 103

43. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

44. The prior art cited is as follows:

45. IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164). Copyright 1993. (Henceforth referred to as "IEEE 1164").

46. Mueller, Martin. "Chronology TimingDesigner V1.2", Printed Circuit Design, San Francisco, CA. January 1993. (Henceforth referred to as "Mueller").

47. Tzartzanis, Nestoras. "Verilog for Behavioral Modeling". February 3, 1998. Reprinted from www-scf.usc.edu/~ee577/tutorial/verilog/verilog_lec.pdf (Henceforth referred to as "Tzartzanis").

48. De Micheli, Giovanni. "Synthesis and Optimization of Digital Circuits", pp.104-106. Copyright 1994.

49. Ralston, A. and Meek, C. "Encyclopedia of Computer Science". Pp.182-183, and 943. Copyright 1976.

50. The claims are subsequently recited for Applicant's convenience. Applicant's attention is also directed to the pertinent sections of the prior art.

51. Claims 5, 9, 10, 11, 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mueller in view of IEEE 1164 and further in view of Ralston.

52. Claims 1-5, 7, 29-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tzartzanis in view of De Micheli.

53. Claims 6, 8, 12-14 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tzartzanis in view of Mueller.

54. Claims 15-18, 20-28, 33-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tzartzanis in view of De Micheli and further in view of Mueller.

55. Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mueller in view of Tzartzanis and further in view of Ralston.

56. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tzartzanis in view of De Micheli and further in view of Ralston.

57. As per Claim 5, Mueller teaches the limitations of Claim 1, as described in ¶21. However, Mueller does not expressly teach the limitation of Claim 5:

> 5. The method of claim 1, wherein providing an indication includes indicating when the released node is in an unknown logic state.

IEEE 1164 does teach (pp.2,4,5) that the node can be in an unknown state.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Mueller with IEEE 1164 because doing so would identify variables "irrelevant to the particular function being represented." (Ralston, p.183)

58. As per Claim 9, Mueller teaches the limitations of Claim 8, as described in ¶27. However, Mueller does not expressly teach the limitation of Claim 9:

> 9. The method of claim 8, wherein the initial node condition is forced again if the testing results in the node resolving to an unknown logic value.

IEEE 1164 does teach (pp.2,4,5) that the node can be in an unknown state. Moreover, the truth tables in IEEE 1164 show the results when an unknown signal and another signal are operated on by an AND or an OR gate. Moreover, IEEE 1164 teaches "Forcing 0" and "Forcing 1" states.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Mueller with IEEE 1164 because doing so identifies variables "irrelevant to the particular function being represented." (Ralston, p.183). It is inherent that forcing these "irrelevant" variables to arbitrary values would have no effect on the particular function being represented.

59. As per Claim 10, Mueller teaches the limitations of Claims 8 and 9, as described in ¶27 and ¶58. However, Mueller does not expressly teach the limitation of Claim 10:

> 10. The method of claim 9, wherein the initial node condition is forced and simulation is repeated until the node resolves to a valid logic value.

IEEE 1164 does teach (pp.2,4,5) that the node can be in an unknown state. Moreover, the truth tables in IEEE 1164 show the results when an unknown signal and another signal are operated on by an AND or an OR gate. Moreover, IEEE 1164 teaches "Forcing 0" and "Forcing 1" states.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Mueller with IEEE 1164 because doing so identifies variables "irrelevant to the particular function being represented." (Ralston, p.183). It is inherent that forcing these "irrelevant" variables to arbitrary values would have no effect on the particular function being represented.

60. As per Claim 11, Mueller teaches the limitations of Claims 8-10, as described in ¶27, ¶58 and ¶59. However, Mueller does not expressly teach the limitation of Claim 11:

> 11. The method of claim 10, wherein monitoring only occurs after the node resolves to a valid logic value.

Mueller does teach (p.2, "Creating Signals") that the node can be in 5 different states, among them, the valid and invalid states. Moreover, Mueller teaches (p.3, "Delays and Constraints", "Miscellaneous") that timing diagrams can created to plot these signals. Thus the monitoring occurs both when the signal is valid and when it is invalid.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to monitor signals (plot them on a timing diagram) only when

they are valid, because doing so would be a decrease in functionality from what is taught by Mueller and what is well known and commonly used in the art.

61. As per Claim 19, Mueller teaches the limitations of Claim 8, as described in ¶27. However, Mueller does not expressly teach the limitation of Claim 19:

> 19. The medium of claim 8, having further computer-executable instructions for indicating when the released node is in an unknown logic state.

IEEE 1164 does teach (pp.2,4,5) that the node can be in an unknown state. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Mueller with IEEE 1164 because doing so identifies variables "irrelevant to the particular function being represented." (Ralston, p.183)

62. Tzartzanis teaches the following limitations of Claim 1:

> 1. A method of simulating a node, comprising:
> forcing an initial logic state on the node;
> (Tzartzanis, p.55)
>
> monitoring the released node; and
> (Tzartzanis, pp.65-66)
>
> providing an indication when the released node is in a preselected condition.
> (Tzartzanis, pp.63-64, p.67, p.70)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

> releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation

warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

63. As per Claim 2, Tzartzanis teaches the limitations of Claim 1, as described in ¶62. Tzartzanis also teaches the limitations of Claim 2:

> 2. The method of claim 1, wherein forcing the initial logic state includes forcing to a logic zero, logic one or high-impedance.
> (Tzartzanis, p.8)

64. As per Claim 3, Tzartzanis teaches the limitations of Claim 1, as described in ¶62. Tzartzanis also teaches the limitations of Claim 3:

> 3. The method of claim 1, wherein releasing the node further comprises determining that the condition is met after passage of a predetermined amount of time.
> (Tzartzanis, p.37, pp.46-50)

65. As per Claim 4, Tzartzanis teaches the limitations of Claims 1 and 3, as described in ¶62 and ¶64. Mueller also teaches the limitations of Claim 4:

4. The method of claim 3, wherein releasing the node further comprises determining that the condition is met when the node has been resolved.
(Tzartzanis, p.37)

66. As per Claim 5, Tzartzanis teaches the limitations of Claim 1, as described in ¶62. Mueller also teaches the limitations of Claim 5:

5. The method of claim 1, wherein providing an indication includes indicating when the released node is in an unknown logic state.
(Tzartzanis, p.8)

67. As per Claim 7, Tzartzanis teaches the limitations of Claims 1 and 3, as described in ¶62 and ¶64. Tzartzanis also teaches the limitations of Claim 7:

7. The method of claim 3, further comprising selecting a user-defined time period for the predetermined amount of time.
(Tzartzanis, p.46-50)

68. Tzartzanis teaches the limitations of Claim 29:

29. An HDL initial condition module comprising means for maintaining a logic level of a simulated circuit node for a predetermined period of time,
(Tzartzanis, pp.35-36, pp.46-50)

means for releasing an initial condition,
(Tzartzanis, pp.28-38)

and wherein the predetermined period of time is a simulation run time defined by an HDL simulation executable program.
(Tzartzanis, pp.35-36, pp.46-50)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation

warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

69. As per Claim 30, Tzartzanis teaches the limitations of Claim 29, as described in ¶68. Tzartzanis also teaches the limitations of Claim 30:

30. The module of claim 29, wherein the predetermined period of time is a user defined period of time.
(Tzartzanis, pp.35-36, pp.46-50)

70. As per Claim 6, Tzartzanis teaches the limitations of Claim 1, as described in ¶62. In regards to Claim 6:

Tzartzanis also teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

6. The method of claim 1, further comprising providing an error indication when the released node is a preselected condition.

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

71. Tzartzanis teaches the following limitations of Claim 8:

8. A method of initializing and monitoring a simulated circuit node, comprising:

obtaining an initial node condition for a node;
(Tzartzanis, p.55)

forcing the node to the initial node condition;
(Tzartzanis, p.55)

simulating a circuit containing the node;
(Tzartzanis, pp.36-50)

testing the node for a valid condition;
(Tzartzanis, p.37)

monitoring the node; and
(Tzartzanis, pp.65-66)

Tzartzanis also teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

providing an indication when the node is in an undesirable condition.

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

72. As per Claim 12, Tzartzanis teaches the limitations of Claim 8, as described in ¶71. Tzartzanis also teaches the limitations of Claim 12:

> 12. The method of claim 8, further comprising outputting the condition of the simulated node.
> (Tzartzanis, pp.63, 67-70)

73. As per Claim 13, Tzartzanis teaches the limitations of Claim 8, as described in ¶71. Tzartzanis also teaches the limitations of Claim 13:

> 13. The method of claim 8, further comprising obtaining a simulation run time.
> (Tzartzanis, pp.44-50)

74. As per Claim 14, Tzartzanis teaches the limitations of Claims 8 and 13, as described in ¶71 and ¶73. Tzartzanis also teaches the limitations of Claim 14:

> 14. The method of claim 13, further comprising outputting a final node condition when the simulation run time is completed.
> (Tzartzanis, pp.63, 67-70)

75. Tzartzanis teaches the limitations of Claim 19:

> 19. The medium of claim 8, having further computer-executable instructions for indicating when the released node is in an unknown logic state.
> (Tzartzanis, p.8, pp.63, 67-70)

76. Tzartzanis teaches the limitations of Claim 15:

> 15. A computer-readable medium having computer-executable instructions comprising:

> forcing an initial logic state on the node;
> (Tzartzanis, p.55)
>
> monitoring the released node; and
> (Tzartzanis, pp.65-66)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

> releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

Tzartzanis also teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

providing an indication when the node is in an undesirable condition.

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

77. As per Claim 16, Tzartzanis teaches the limitations of Claim 15, as described in ¶76. Tzartzanis also teaches the limitations of Claim 16:

16. The medium of claim 15, having further computer-executable instructions for forcing the initial logic state to a logic zero, logic one or high-impedance.
(Tzartzanis, p.8)

78. As per Claim 17, Tzartzanis teaches the limitations of Claim 15, as described in ¶76. Tzartzanis also teaches the limitations of Claim 17:

17. The medium of claim 15, having further computer-executable instructions for determining that the condition is met after passage of a predetermined amount of time.
(Tzartzanis, pp.46-50)

79. As per Claim 18, Tzartzanis teaches the limitations of Claim 15, as described in ¶76. Tzartzanis also teaches the limitations of Claim 18:

18. The medium of claim 15, having further computer-executable instructions for determining that the condition is met when the node has been resolved (Tzartzanis, pp.37)

80. Tzartzanis teaches the limitations of Claim 20:

20. A simulation module for initializing and monitoring a simulated circuit node, comprising:

an input means for inputting an initial node condition;
(Tzartzanis, p.55)

a conveying means for conveying the initial node condition to a simulated node;
(Tzartzanis, p.55)

a monitoring means for monitoring the simulated node for a node condition;
(Tzartzanis, pp.65-66)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be

described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

In addition, Tzartzanis teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

> and an output means for outputting an indication when the node condition is in an undesirable state.

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

81. As per Claim 21, Tzartzanis teaches the limitations of Claim 20, as described in ¶80. Tzartzanis also teaches the limitations of Claim 21:

> 21. The module of claim 20, further comprising an output means for outputting the node condition.
> (Tzartzanis, pp.63, 67-70)

82. As per Claim 22, Tzartzanis teaches the limitations of Claim 20, as described in ¶80. Tzartzanis also teaches the limitations of Claim 19:

22. The module of claim 20, further comprising an input means for inputting a simulation run time.
(Tzartzanis, p.60)

83. As per Claim 23, Tzartzanis teaches the limitations of Claim 22, as described in ¶82. Tzartzanis also teaches the limitations of Claim 23:

23. The module of claim 22, further comprising an output means for outputting a final node condition at completion of the simulation run time.
(Mueller, p.3, "Miscellaneous")

84. Tzartzanis teaches the limitations of Claim 24:

24. A computerized system for initializing and monitoring a simulated circuit node, the system comprising:

a circuit simulation tool;
(Tzartzanis, p.3)

a first input module inputting an initial node condition;
(Tzartzanis, p.55)

a conveying module conveying the initial node condition to a simulated node;
(Tzartzanis, pp.58-62)

a monitoring module monitoring the simulated node for a node condition;
(Tzartzanis, pp.58-62, 65-66)

a second input module inputting a simulation run time; and
(Tzartzanis, p.55 and pp.58-62)

a second output module outputting a final node condition at completion of the simulation run time.
(Tzartzanis, pp.58-62, 63, 67-70)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation

warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

In addition, Tzartzanis teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

> a first output module outputting an indication when the node condition is in an undesirable state;

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of

Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

85. Tzartzanis teaches the limitations of Claim 25:

> 25. An HDL initial condition module comprising a means for maintaining a logic level of a simulated circuit node until a release condition is met.
> (Tzartzanis, p.37)

86. As per Claim 26, Tzartzanis teaches the limitations of Claim 25, as described in ¶85. Tzartzanis also teaches the limitations of Claim 26:

> 26. The module of claim 25 wherein the release condition is when the node can be resolved to a known logic state.
> (Tzartzanis, p.8)

87. As per Claim 27, Tzartzanis teaches the limitations of Claim 25, as described in ¶85. Tzartzanis also teaches the limitations of Claim 27:

> 27. The module of claim 25 wherein the logic level is a value defined by an HDL executable simulation program.
> (Tzartzanis, pp.3, and 8)

Verilog is an HDL executable simulation language.

88. Tzartzanis teaches the limitations of Claim 28:

> 28. An HDL initial condition module having an initial condition release means and a simulated circuit node error detection means.
> (Tzartzanis, pp.28-36, pp.65 - pp.70)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

> releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation

warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

In addition, Tzartzanis teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.
However, Tzartzanis does not expressly teach the following limitation:

a simulated circuit node error detection means

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of

Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

89. Tzartzanis teaches the limitations of Claim 33:

33. An HDL design tool, comprising:
a circuit simulation device; and
(Tzartaris, p.3)

a plurality of selectable modules capable of being linked to the circuit simulation device,
(Tzartaris, p.51-60)

wherein at least one of the selectable modules executes the following commands:

inputting an initial node condition;
(Tzartaris, p.55)

conveying the initial node condition to a simulated node;
(Tzartaris, p.55)

monitoring the simulated node for a node condition; and
(Tzartaris, pp.65-66)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

Tzartzanis also teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

> an output means for outputting an indication when the node condition is in an undesirable state.

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

90. Tzartzanis teaches the limitations of Claim 34:

34. A simulation method, comprising:
phase one, including;
forcing an initial logic zero, logic one or high-impedance on a node;
(Tzartzanis, p.8 and p.55)

testing to see if the node has been resolved;
(Tzartzanis, p.37)

if the node has been resolved, continue to phase two
(Tzartzanis, p.52, pp.60-61)

if the node has not been resolved, continuing in phase one
phase two, including;
(Tzartzanis, p.37)

monitoring the node value;
(Tzartzanis, pp.65-66)

testing the node value;
(Tzartzanis, p.37)

and, continuing in phase two until simulation completion.
(Tzartzanis, pp.58-62)

Moreover, Tzartzanis teaches (Page 37) an if-then-else construct in the Verilog circuit simulation software language.

However, Tzartzanis does not expressly teach:

releasing the node if a predetermined condition is met and creating therefrom a released node;

Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

Tzartzanis also teaches the display of the simulation output values on a computer screen (p.63), in a file (p.67), and as a graph on a computer screen (p.70). These are all indications of the condition of a node.

However, Tzartzanis does not expressly teach the following limitation:

indicating an error if an unacceptable condition appears on the node;

Mueller does expressly teach "if a constraint is violated, its markers and labels, including the calculated margin, will turn red." (Mueller, p.3, "Delays and Constraints").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of Mueller by changing the color of the output for undesirable conditions, because doing so shows the undesirable conditions more vividly.

91. Tzartzanis teaches the limitations of Claim 35:

35. The method of claim 34, wherein simulation completion is a user defined time period.
(Tzartzanis, pp.46-50)

92. In regards to the limitations of Claim 31:

31. An HDL simulated circuit device, comprising:
a first HDL module comprising:
a first input submodule inputting a first initial node condition;
a first conveyance submodule conveying the first initial node condition to a first simulated node;
a first monitor submodule monitoring the first simulated node for a first node condition; and
a first output submodule outputting a first indication when the first node condition is in an undesirable state;
a second HDL module comprising:
a second input submodule inputting a second initial node condition;
a second conveyance submodule conveying the second initial node condition to a second simulated node;
a release submodule releasing the node on a predetermined condition;
a second monitor submodule monitoring the second simulated node for a second node condition; and
a second output submodule outputting a second indication when the second node condition is in an undesirable state; and

wherein the first conveyance submodule additionally conveys the first initial node condition to the second input submodule.

Mueller teaches inputting an initial node condition (p.2, "Creating Signals"), monitoring the node (p.3, "Miscellaneous"), and outputting indications when the node is in an undesirable state (p.3, "Miscellaneous).

However, Mueller does not expressly teach that his product is implemented in HDL. Moreover, Mueller does not expressly teach the use of "submodules".

Tzartzanis does teach implementing the above features in HDL. (pp.28-38, pp.65-70). Moreover, Tzartzanis teaches the use of modular blocks (pp.51-55, 58-62).

At the time the invention was made, it would have been obvious to one of ordinary skill in the art to modify the teachings of Mueller with Tzartzanis in order

to implement the features in and HDL language such as Verilog, because this was well known and commonly used in the art.

Moreover, it would have been obvious to one of ordinary skill in the art to modify the teachings of Mueller with Tzartzanis in order to implement the features in a modular program, because this would make it "simpler to design, write, and test" (Ralston, p.943).

93. In regards to the limitations of Claim 32:

32. An HDL simulated circuit device, comprising:
a first HDL module comprising:
a first input;
a first conveyance;
a first node condition output
a second HDL module comprising:
a second input;
a second conveyance;
a third HDL module comprising:
a release condition;
...
wherein the first node condition output means outputs the first node condition to the second input means if the release condition is valid.

Tzartzanis teaches the use of Verilog, an HDL language. Moreover, he teaches inputs (p.35, p.55), conveyance (p.37), outputs (pp.38-39), and release conditions (p.37). Moreover, Tzartzanis also teaches Block statements and modules (pp.58-62).

However, Tzartzanis does not expressly teach a "release condition". Applicant defines the "release" of a node (specification, p.5) as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced."

De Micheli (Example 3.2.5, p.105) teaches the use of an IF-THEN-ELSE construct in the VHDL circuit simulation language to implement the "release" functionality as defined and claimed by the applicant.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Tzartzanis with the teachings of De Micheli because, as De Micheli teaches, "finite state machines can be described by procedural models where a variable keeps the state information. Then, the operations of the finite-state machine can be described as an iteration (synchronized to the clock), with branching to the fragments corresponding to the present state."

Moreover, Tzartzanis does not expressly teach exactly three related HDL modules.

Ralston (p.943) teaches that "A program consisting of modules of properly designed scope ... is much simpler to design, write, and test than is the same program when it is not so modularized.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to make three related modules that perform these functions, because that is a matter of design choice, and doing so would make the program "simpler to design, write, and test" (Ralston, p.943).

See MPEP §2144.04, Section VI, and *In re Harza*, 274 F.2d 669, 124 USPQ 378 (CCPA 1960), *In re Japikse*, 181 F.2d 1019, 86 USPQ 70 (CCPA 1950), and *In re Kuhle*, 526 F.2d 553, 188 USPQ 7 (CCPA 1975).

Response to Arguments

94. Applicants' arguments filed 8/12/99 have been fully considered but they are not persuasive.

Claim Objections

95. Examiner acknowledges Applicant's amendments to Claims 6 and 19. The objections to these Claims have been withdrawn.

Claim Rejections - 35 USC § 102

96. Applicant argues (paper #5, p.3) that Mueller does not teach a "release" element, which is claimed in Claims 1, 15, 20 and 24. For the sake of clarity, the Examiner has provided the following clarifications.

Applicant is reminded of Specification, p.5, in which Applicant defines the "release" of a node as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced." This was cited in the Claim Interpretations section of Paper #4.

Moreover, Applicant is referred to the "CREATING SIGNALS" section of Mueller, which teaches: "Every time another edge is added to a waveform, the state will change according to a toggle pattern determined by the initial state of the system" and "Every signal can be renamed, moved, deleted and locked into place for easy reference when scrolling through a large number of signals. Any

signal edge can be redefined by clicking on it and either specifying a different signal level or dragging it to another point in time."

The signal is clearly forced by the user, until such time when the user decides that the simulation warrants a change in the logic value. Therefore, Examiner finds that the functionality of "releasing" a node, as defined by Applicant in the specification, is taught by Mueller. The rejections of claims 1, 15, 20, 24, and dependent claims 2-4, 6-7, 16-18, and 21-23 are maintained.

97. Applicant argues (paper #5, p.3) that the first office action "fails to state a prima facie case of anticipation" with respect to claims 2-4, 6-7, 16-18, and 21-23. Applicant is referred to the rejections of these claims, based on Mueller, in ¶22-¶26, ¶32-¶34, and ¶36-¶38.

98. Applicant argues (paper #5, p.3-4) that Mueller does not teach a "... providing an indication when the node is in an undesirable condition", which is claimed in Claim 8. For the sake of clarity, the Examiner has provided the following clarifications.

Applicant is referred to the "CREATING SIGNALS" section of Mueller, which teaches: "Each new signal to be added requires specifying an alphanumeric name and one of five initial states: high (H), low (L), high-impedance (Z), valid (V), and invalid (I)" and "Every time another edge is added to a waveform, the state will change according to a toggle pattern determined by the initial state of the system".

Examiner finds the "invalid (I)" state as being equivalent to the applicants claimed "undesirable" condition. The rejections of claim 8 and dependant claims 12-14 are maintained.

99. Applicant argues (paper #5, p.4) that the first office action "fails to state a prima facie case of anticipation" with respect to claims 12-14. Applicant is referred to the rejections of these claims, based on Mueller, in ¶28-¶30.

100. Applicant argues (paper #5, p.4) that IEEE 1164, pp.2-4, does not teach "... a means for maintaining a logic level of a simulated circuit node until a release condition is met", which is claimed in Claim 25. This is mere attorney argument given that the Applicant has not provided an explanation as to how the limitations differ from the cited prior art. For the sake of clarity, the Examiner has provided the following clarifications.

Applicant's attention is also drawn to the fact that the rejection was based on pp.2-**5** of the IEEE reference, and not pp.2-**4** as stated by the Applicant (paper #5, p.4).

Applicant is reminded of Specification, p.5, in which Applicant defines the "release" of a node as meaning that "the simulation program is free to change the logic value of the node if [the] simulation warrants a change in the logic value. Before the release, the logic value of the node is forced." This was cited in the Claim Interpretations section of Paper #4.

Applicant is referred to the "truth table for 'and' function" on p.5 of the IEEE reference, which teaches that for signal levels (except "0" and "L") are

maintained until they undergo an "and" operation. This is functionally equivalent to Applicant's definition of a "release" condition.

The rejections of claim 25 and dependant claims 26-27 are maintained.

101. Applicant argues (paper #5, p.4) that the first office action "fails to state a prima facie case of anticipation" with respect to claims 26-27. Applicant is referred to the rejections of these claims, based on IEEE 1164, in ¶41-¶42.

102. Applicant argues (paper #5, pp.4-5) that p.37 of Tzartzanis does not teach a "release" element, which is claimed in Claims 1, 15, 20, 24, 25, 28, 29, 33, and 34. These rejections, and the rejections of the dependant claims, have been replaced with 35 USC 103 rejections, with the claims being unpatentable over Tzartzanis in view of De Micheli.

103. Applicant argues (paper #5, pp.5-6) that pp.63 and 67-70 in Tzartzanis does not teach "... providing an indication when the node is in an undesirable condition ...", which is claimed in claim 8. For the sake of clarity, the Examiner has provided the following clarifications. This rejection has been replaced with 35 USC 103 rejections as being unpatentable over Tzartzanis in view of Mueller.

104. Applicant argues (paper #5, p.6) that the first office action "fails to state a prima facie case of anticipation" with respect to claims 12-14. Applicant is referred to the rejections of these claims, based on Tzartzanis in view of Mueller, in ¶72-¶74.

Claim Rejections - 35 USC § 103

105. Applicant argues (paper #5, pp.5-6) in regards to Claims 5, 9, 10, 11, and 19 that "the statements in paragraphs 86-90 related to increasing the resolution of the simulation, reducing ambiguity in the simulation results and decreasing the functionality of the system do not constitute specific, objective evidence of record. They are simply not found in the record." The motivation statements of Claims 5, 9, 10, 11 and 19 have been modified.

106. Applicant argues (paper #5, p.7) in regards to Claim 31 that the statements in paragraph 92 fail "to provide specific, objective evidence of record for a finding of a suggestion or motivation to combine the reference teachings". The motivation statement of Claim 31 has been modified to reflect the teachings of Ralston, p.943.

107. Applicant argues (paper #5, pp.7-8) in regards to Claim 32 that the "release condition" is not taught by Tzartzanis. The release condition is taught by De Micheli. Moreover, Applicant argues that Tzartzanis does not expressly teach three related modules. Ralston (p.943) teaches the use of modular programming, and the use of three modules is a matter of design choice (see See MPEP §2144.04 part VI). Claim 32 has been modified to represent these changes.

Conclusion

108. The following prior art, made of record and not relied upon, is considered pertinent to applicant's disclosure.

109. Silicon Integration Initiative, Inc. "Electronic Component Information eXchange (ECIX) – Timing Diagram Markup Language (TDML) Sample Instances". May 18, 1999.

This document has a screen shot of a Timing Diagram produced by the software program TimingDesigner. This screen shot shows that a Timing Diagram provides an indication when a signal is in a "pre-selected" condition.

The user can interpret this indication as an "error indication".

110. IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164). Copyright 1993. The IEEE Standard expressly teaches (pp.2,4,5) an "unknown state".

111. McDermott, Robert. "Transmission Gate Modeling in an Existing Three-Value Simulator". 19th IEEE Design Automation Conference. 1982.

112. U.S. Patent 5,257,363.

113. U.S. Patent 6,421,808.

114. U.S. Patent 5,980,096.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ayal I. Sharon whose telephone number is (703) 306-0297. The examiner can normally be reached on Monday through Thursday, and the first Friday of a biweek, 8:30 am – 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska can be reached on (703) 305-9704. Any response to this office action should be mailed to:

Director of Patents and Trademarks
Washington, DC 20231

Hand-delivered responses should be brought to the following office:

4th floor receptionist's office
Crystal Park 2
2121 Crystal Drive
Arlington, VA

The fax phone numbers for the organization where this application or proceeding is assigned are:

Official communications:	(703) 746-7239
Non-Official / Draft communications	(703) 746-7240
After Final communications	(703) 746-7238

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist, whose telephone number is: (703) 305-3900.

Ayal I. Sharon

Art Unit 2123

October 29, 2002

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER